Patent

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

TITLE OF THE INVENTION

METHOD AND APPARATUS FOR SERVER LOAD BALANCING

INVENTORS

MICHAEL YIP
DESIKAN SARAVANAN
ARTHUR LIN
ED ROVNER
TASH HEPTING
PAUL ANDERSON
BRIAN BAILEY

Prepared by

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(503) 684-6200

Express Mail Label No.: EL034438952US

Attorney Docket No.: 002717.P017

COPYRIGHT NOTICE

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention is related to a method and apparatus for controlling traffic in an internetwork. In particular, the present invention is related to a method and apparatus for server load balancing based on a hashing function.

Description of the Related Art

The growth of the Internet has caused problems unanticipated in its original design. As new users (clients) connect to the Internet. For example, to access servers for e-commerce, financial transactions, and intranet applications, the load on the servers has greatly increased. As the number of clients that access a server increase, the number of data packets processed by the server also increases. This increased load on the server may result in communications between clients and the server slowing down or timing out.

To solve this problem, the processing capability of the server may be increased, or a server with increased processing capacity may replace the old server. Ultimately, as shown in Figure 1, additional servers 106-108, may be added to form a cluster of servers or server bank 110 to respond to the needs of clients. The server bank 110 provides a

plurality of servers, each with the same content. Users, therefore, may connect to any one of the servers to access desired information. As can be seen, however, it is desirable to balance the demand, or load, placed on any one server, so that the resources of the server bank are effectively utilized. To that end, a Server Load Balancing (SLB) algorithm may be employed at router 105 wherein sessions from any one of clients 101-n are connected to a particular server in the server bank. The term clients, as used herein is construed to mean personal computers, wireless devices etc. that may be used to communicate with a server. As examples, a round robin algorithm, or a least connections algorithm is employed by a SLB algorithm to direct communications to one of a plurality of servers in the server bank.

With reference to Figure 2, a SLB address table 200 may be maintained in router 105 to facilitate connections between clients and a server in the server bank. The SLB address table maintains information, in particular, address information for each server in the server bank, and each client that accesses the server bank, and maintains a mapping or binding between a particular client and server. This facilitates maintaining "sticky connections" between the server and client. A sticky connection is a connection wherein the same client communicates with the same server across multiple connection requests. For example, during multiple Transmission Control Protocol (TCP) (see RFC 793 and 879 for details on TCP) connection requests, a user accesses multiple related web pages during the filling out of a multi-page form using HyperText Transfer Protocol (HTTP) (see RFC 1945 for details on HTTP). The SLB address table contains an entry for each TCP connection comprising the Internet Protocol (IP) address of the client (source), and the server (destination), and the TCP port numbers of the client and server. When a data

packet arrives at router 105 from a client, the router looks up in the SLB address table for an entry for a connection between the client and a server in the server bank. If there is a connection, the data packet is forwarded to the server in the server bank identified by the server IP address specified in the entry. However, if an entry does not exist in the SLB address table, a server is mapped to the client based on, for example, a round robin or least connections algorithm, and an entry corresponding to the connection is made in the SLB address table.

Subsequent data packets from a particular client connection are routed to the allocated server in the server bank by the router after performing a look up in the SLB address table as disclosed above. With numerous data packets from different clients, the router looks up the appropriate server's IP address in the SLB address table for each data packet received. With a large number of client connections, the size of the SLB address table can become very large, and a considerable amount of processing overhead is incurred by the router in searching the table for the address of the server prior to forwarding each data packet. As shown in Figure 2, for 'n' clients, the router may have to search through at least 'n' rows of data for a server's IP address. This process is time consuming, inefficient and may require considerable memory to store the SLB address table. What is needed, therefore, is a more efficient way to balance traffic load among a plurality of servers.

BRIEF SUMMARY OF THE INVENTION

The invention discloses receiving a data packet at a router, generating a hashed value based at least in part on the source and the destination address specified in the data packet, selecting a server in a server bank based on the hashed value, and forwarding the data packet to the selected server.

BRIEF SUMMARY OF THE DRAWINGS

Examples of the present invention are illustrated in the accompanying drawings. The accompanying drawings, however, do not limit the scope of the present invention. Similar references in the drawings indicate similar elements.

Figure 1 provides a depiction of the environment in which the invention is used.

Figure 2 illustrates an example of an SLB address table that is used in the prior art.

Figure 3 illustrates an example of a server address table in accordance with an embodiment of the invention.

Figure 4 is a flow diagram of an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

A method and apparatus is described for server load balancing. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one of ordinary skill in the art that the present invention may be practiced without these specific details. In other instances, well-known architectures, steps, and techniques have not been shown to avoid unnecessarily obscuring the present invention. For example, specific details are not provided as to whether the method is implemented as a software routine, hardware circuit, firmware, or a combination thereof.

Parts of the description will be presented using terminology commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. Also, parts of the description will be presented in terms of operations performed through the execution of programming instructions. As well understood by those skilled in the art, these operations often take the form of signals capable of being stored, transferred, combined, and otherwise manipulated through, for instance, electrical components.

Various operations will be described as multiple discrete steps performed in turn in a manner that is helpful in understanding the present invention. However, the order of description should not be construed as to imply that these operations are necessarily performed in the order they are presented, or even order dependent. Lastly, repeated usage of the phrase "in one embodiment" does not necessarily refer to the same embodiment, although it may.

In Figure 1, an example internetwork 120 in which an embodiment of the present invention may be utilized is illustrated. Data packets flow between clients 101-n and servers 106-n via router 105. Router 105 is between servers in the server bank 110 and internetwork 120. Although the embodiment described herein utilizes a router, other embodiments may utilize any device to control the flow of data packets between clients and servers. For e.g., switches, computing devices, wireless devices etc. The term client as used here is generic and is intended to mean any device, including but not limited to, a computing device, router, switch, and gateway that is the source of one or more communications with a server. So also the term server, as used here is generic and is intended to mean any device that communicates with a client, including a firewall SSL accelerator, primarily to provide data or programming content to the client. While the description that follows addresses the method as it applies to an internetwork (internet) architecture, it is appreciated by those of ordinary skill in the art that the method is generally applicable to any network architecture including, but not limited to, Local Area Networks (LANs), Metropolitan Area Networks (MANs), and Wide Area Networks (WANs).

In one embodiment, a web browser on a client accesses a HyperText Transfer Protocol (HTTP) based server hosting one or more websites on the World Wide Web (www). When a data packet transmitted from client 101 arrives at router 105, the source address and the destination address are obtained from the data packet. A hashing function is performed using the addresses as input, to obtain a hashed value.

In other embodiments, it may be possible to use HTTP or other "layer 5" or higher layer information in the packet header to obtain a unique key as input to the

hashing function. In still other embodiments, parts of the TCP port numbers or parts of the IP addresses may be used as input to the hashing function.

In one embodiment, the hashed value obtained from the hashing function is divided by the number of entries in a server load sharing table as illustrated in Figure 3. The server load sharing table contains either the IP address or the MAC address of the server or the switch port where the server is located. The router then uses the information in the server load sharing table to forward the data packet.

When subsequent data packets from the same client 101 arrive at router 105, a hashed value is obtained using the hashing function as described above. Using the hashed value as an index to the server load sharing table 300, the information of the server is obtained and the subsequent data packets are transmitted to the same server out the port number specified by the router's forwarding database. For the example shown in Figure 3, the SLB algorithm in one embodiment of the invention searches, at most, n rows of the server address table, wherein n represents the number of servers in the server bank. The disclosed invention thereby eliminates the need for the SLB address table and eliminates lengthy and time consuming searches for a server's address.

Figure 4 illustrates a flow diagram of the process of receiving a data packet at router 105, and forwarding the data packet to a server in server bank 110. At 405, router 105 receives a data packet from a client. At 415, a hash is performed using the IP addresses and the TCP port number as described earlier. At 420, the router searches the server address table 300, using the hash value obtained as an index, for the IP address of a server in the server bank to which the data packet is to be forwarded. At 425, the router looks up the physical port out which to forward the data packet in the router's forwarding

database, and at 430, the data packet is forwarded to the server in the server bank via the port number determined above. Subsequent data packets from the same client are forwarded to the same server by repeating the process described above.

Embodiments of the invention, including, but not limited to, tables utilized by the invention, may be represented as a software product stored on a machine-accessible medium (also referred to as a computer-accessible medium or a processor-accessible medium). The machine-accessible medium may be any type of magnetic, optical, or electrical storage medium including a diskette, CD-ROM, memory device (volatile or non-volatile), or similar storage mechanism. The machine-accessible medium may contain various sets of instructions, code sequences, configuration information, or other data. For example, the procedures described herein for controlling the flow of data packets between clients and servers, can be stored on the machine-readable medium. Those of ordinary skill in the art will appreciate that other instructions and operations necessary to implement the described invention may also be stored on the machine-accessible medium.